UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

MICHAEL PHILIP KAUFMAN,

Plaintiff,

v.

MICROSOFT CORPORATION,

Defendant.

No. 16-CV-2880-LTS-SN

JURY TRIAL DEMANDED

PLAINTIFF'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

PAGE(S)

Cases

INTRODUCTION

U.S. Patent No. 7,885,981 (the "'981 Patent") to Michael Philip Kaufman ("Kaufman") and Micah Philip Silverman, discloses and claims systems and methods for generating an automatic user interface for arbitrarily complex or large databases, and specifically for relational databases, which are collections of data organized into a set of related tables. The tables are related in that one table might contain details/attributes for a list of items that are in a second table (*e.g.*, a table of departments organized by Department Number and a related table of employees, including their respective Department Numbers).

Plaintiff Kaufman, who holds full title to the '981 Patent, accuses Defendant Microsoft Corporation ("Microsoft"), and specifically Microsoft's Dynamic Data product, of infringing claims 1-5 of the '981 Patent (the "Asserted Claims").

Technology for performing some individual aspects of the patent claims existed before the conception of the claimed subject matter. For example, there were pre-existing user interfaces for viewing individual database tables, with some automated functions. But the '981 Patent discloses and claims taking a pre-existing database and automatically generating a fully-functional user application for working with the database. The claimed subject matter allows for automatically creating an application that can view database tables, navigate between related data elements in separate database tables, and add and edit content within the database in the course of viewing and navigating. During these operations, the application constrains both inputs and displays in order to reflect and maintain the referential integrity of the database, *i.e.*, constrains data inputs, updates, and deletes so as not to violate the definitions of table-to-table relationships within the database. The specification further includes programming source code

that provides working examples of various aspects of the disclosed subject matter, including all aspects of the Asserted Claims.

The parties dispute a limited set of terms from the Asserted Claims of the '981 Patent (the "Disputed Terms"), as reflected in the Joint Claim Terms Chart (D.I. 36). In this brief, Kaufman will address the construction of the Disputed Terms.[1]

BACKGROUND

1. Technological Overview

A database is a collection of data, organized in certain ways, generally to optimize efficiency of search and retrieval. A relational database is the most common type of database, in which data is broken into separate tables that are related to one another. Each table contains conceptual columns and rows. Below is a simple example for a table we will call "Department":

Department_ID	Department Name	Charge Code
100	Aerospace	291743
200	Chemistry	327265

In this example, there are three columns, or "fields" with the column names Department_ID, Department Name, and Charge Code. There are also two rows, or "records" in the table, one for the Aerospace department and one for the Chemistry department.

[1] In addition, Microsoft's proposals suggest that it may challenge a number of terms in the Asserted Claims as indefinite, despite their clear meaning when read from the perspective of a person of ordinary skill in the art in light of the specification and in accordance with the applicable legal standards. If and when Microsoft makes indefiniteness arguments in its responding brief, Plaintiff will respond to such indefiniteness arguments in his reply brief.

Relational databases generally have a number of data tables that are related to one another. Following the example above, there might be another table called "Employee":

Employee_ID	First Name	Last Name	Department_ID
1	Orville	Wright	100
2	Marie	Curie	200

Here we have four columns, with the column names Employee_ID, First Name, Last Name, and Department_ID. There are also two rows, one for Orville Wright and one for Marie Curie.

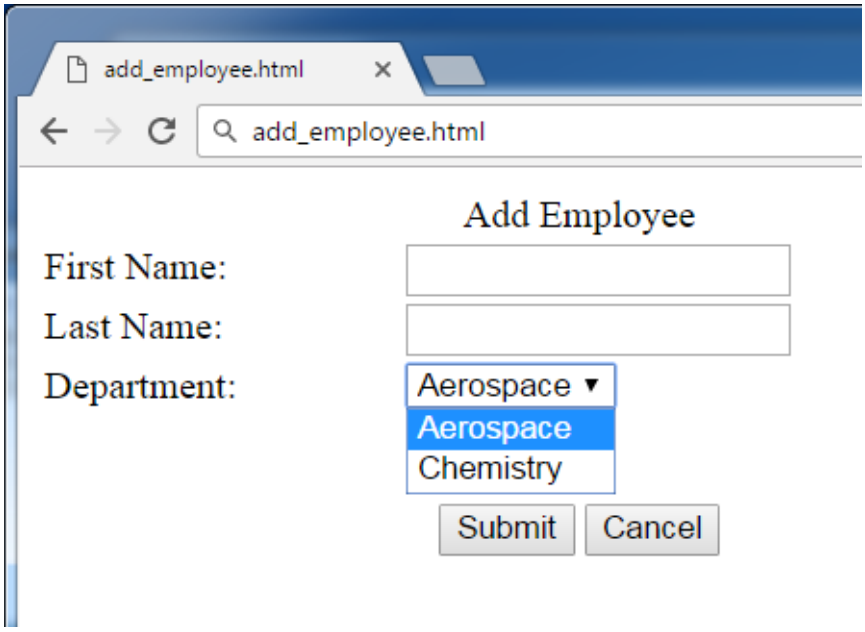
In order to associate employees with their respective departments, the Department_ID column in the Employee table links each employee row to its corresponding department record (having the same Department_ID) within the Department table. Assuming Mr. Wright was in the Aerospace department, the database entry for Mr. Wright in the Employee table would link to the Aerospace department entry in the Department table through its Department_ID (100).

Whether an application is on your computer or on a web site, most of the logic to run a computer application is stored as a program outside of the database itself. The program will run and display menus, graphics, data (and other display aspects), but will "query" the database for the needed data. Since all data for all users of a program is stored in one central repository, data changes made by one user will be visible to all users (*e.g.*, Orville Wright decides to legally change his name to Darrell).

There are various methods of querying a database, and adding, editing, and deleting data from it. All of those methods involve programs ("applications") that access database data. In the prior art, in order to create an application for working with the data in a database, programmers

needed to manually code the data manipulation programs. In order to write those programs, the programmers needed to review the database structure (to get the names, sizes, data types, and other attributes of the tables) and manually code user interfaces, for each database, based on the table and column names and attributes defined in that database. In order for the application to observe and enforce relationships among tables as defined in the database, both to visualize the relationships between data in related tables, and to maintain the integrity of the database when the user adds, updates, or deletes data in the database (for example, so as not to allow adding to an employee record a department that does not exist in the Department table), it was necessary for programmers to manually code all of the necessary logic.

For the example above, a programmer wishing to write an add page for the Employee table would have to look at the number, names, and types of columns to be displayed to the end user to allow for a simple add page to be displayed. The programmer would write the code to render a display such as the following:



The required coding of the functionality behind the above example (including, *inter alia*, listing all the fields to be added, defining labels for them, and their layout), as well as all other

pages that would be required to interact with the database, is eliminated by the claimed subject matter, which automatically creates all of the pages and functionality for working with the database.

2. Claims

The Asserted Claims recite features of an automated solution for what was previously a tedious and error-prone manual process of application creation. The process may start with an existing database without any application, and the claimed subject matter will automatically generate an application adapted for the database and equipped to perform all essential operations that the application needs to perform with the database, including the ability to create, retrieve, update, and delete data in the database, while observing and enforcing the inter-table relationships defined in the database (and thus maintaining the integrity of the database when data is added, updated, or deleted).

In order to generate a client application so adapted to the database, the systems and methods that generate the application need to have detailed information about the database. This information is stored in the database "schema" (or data dictionary). The systems and methods interrogate the database schema to determine structural information about the database, and use that information in constructing a client application having the above-described data-manipulation capabilities.

The automatically generated client application provides a connection to the database. The user interface provided by the application includes a set of modes for interacting with a given database table, including create, retrieve, update and delete, ("CRUD"), and a corresponding display format for each mode. The application further provides displays of the data within the database, table-by-table, in accordance with the display formats. The display formats provide

mechanisms for representing, navigating, and managing data relationships across tables, for selecting among said modes, and for interacting, in accordance the selected mode, with the data in the tables that are reached by said navigation. All of this is done while observing and enforcing relational interdependencies among data across said tables.

CLAIM CONSTRUCTION PRINCIPLES

Construction of patent claims is a matter of law for the Court. *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 391 (1996). This is true even where the construction issues have evidentiary underpinnings in which the District Court must make factual determinations. *Id.* at 390.

"It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*) (internal quotations omitted).

In construing a term, the "objective baseline" is the "ordinary and customary meaning," which is "the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Id.* at 1312-13. "[T]he person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification" and the prosecution history. *Id.* at 1313.

"[T]he best source for understanding a technical term" is a patent's intrinsic evidence, which includes the patent and its prosecution history. *Id.* at 1315 (internal quotations omitted). The specification and the prosecution history "provide[] evidence of how the PTO and the inventor understood the patent." *Id.* at 1317. However, because the prosecution history represents

an "ongoing negotiation . . . rather than the final product of that negotiation, it often lacks the clarity of the specification and thus is less useful for claim construction purposes." *Id.*

"Extrinsic evidence," such as expert and inventor testimony, dictionaries, treatises, and other evidence external to the patent and prosecution history may also be considered in claim construction. *Id.* Extrinsic evidence, however is "less significant than the intrinsic record in determining the legally operative meaning of claim language." *Id.* (internal quotations omitted). A court may rely on extrinsic evidence "in order to better understand the underlying technology and may also rely on dictionary definitions when construing claim terms, so long as the dictionary definition does not contradict any definition found in or ascertained by a reading of the patent documents." *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1584 n.6 (Fed. Cir. 1996).

A further relevant canon of construction concerns whether a claim can be interpreted in a manner that would exclude from its scope the preferred embodiment of an invention as described in the specification. The applicable rule in this regard is that "a claim construction that excludes a preferred embodiment . . . is rarely, if ever correct and would require highly persuasive evidentiary support." *Epos Techs. Ltd. v. Pegasus Techs. Ltd.*, 766 F.3d 1338, 1347 (Fed. Cir. 2014) (*quoting Anchor Wall Sys., Inc. v. Rockwood Retaining Walls, Inc.*, 340 F.3d 1298, 1308 (Fed. Cir. 2003)); *Vitronics*, 90 F.3d at 1583.[2]

[2] Note that "when there is a clear disclaimer during the prosecution history, it may be appropriate to read out the preferred embodiments." *Revolution Eyewear, Inc. v. Aspex Eyewear, Inc.*, 175 F. App'x 350, 356 (Fed. Cir. 2006).

DISPUTED CONSTRUCTIONS

1. "wherein said relational database may be of any arbitrary size or complexity" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
The schema, rather than the data within the database, may be of any arbitrary size or complexity	The claim term lacks support in the specification, otherwise, plain and ordinary meaning

Kaufman has requested construction of this term to avoid confusion and unnecessary argument later in the case. Kaufman believes that the plain meaning of this term makes clear that "arbitrary size or complexity" refers to the database schema, rather than the data stored within the database. The database schema is the organizational structure of the database, and encompasses the tables, fields, and relationships of those tables and fields. Merely adding data to a database does not make it more "complex." The complexity arises as a result of having numerous tables defined in the database schema (in this sense, the "size" of the schema), and in the number and interrelationships of the tables, also defined in the schema.[3]

The first words in the "Brief Summary of the Invention" section of the specification state that "It is an object of the invention to provide a complete and fully functional user interface (UI) *for any arbitrarily complex or large database schema*, without any custom software programming." ('981 Patent at 3:8-11 (emphasis added).) These words make clear what the phrase "arbitrary size or complexity" is modifying, *i.e.*, the database schema. The claimed

[3] Microsoft asserts that the claim provision that the "relational database may be of any arbitrary size or complexity" lacks support in the specification. This is untrue. This claimed feature is explicitly discussed throughout the specification, as will be discussed herein.

subject matter is extensible to any relational database, regardless of the "size or complexity" of its schema. Further in the summary, the specification explicitly links the word "complex" with the database schema, *i.e.*, the tables, constraints, and relationships. "This embodiment has the capability of creating such a UI where the underlying RDBMS is complex and comprises a plurality of tables, constraints, and relationships." ('981 Patent at 3:47-50.)

In distinguishing pre-existing tools to ease some of the burden of database UI development, "considerable work remains in building a complete, fully-functional UI for a *back-end schema of any appreciable size or complexity*." ('981 Patent at 2:65-67 (emphasis added).)

When discussing the preferred embodiment, as provided in the disclosed source code, the specification discusses a user interface "to any arbitrarily large or *complex relational database schema* (or 'data model')." ('981 Patent at 4:51-55 (emphasis added).) The disclosed source code in the specification supports schemas of arbitrary size and complexity, by allowing an unbounded number of traversals between tables, where each traversal goes from a table to a foreign-key table, or vice versa.

The file history of the '981 Patent further supports the interpretation that the phrase "arbitrary size or complexity" concerns the schema, or structure, of the database. During the prosecution of U.S. Patent No. 7,318,066, the patent to which the '981 Patent claims priority, Kaufman distinguished the claims from a piece of prior art, Hecht, by arguing that the claims of the parent case "involve[] a system for giving a user access, through a user interface, to the data contained within an arbitrary existing database *structure*." (U.S. Patent Application No. 10/428,209, March 14, 2005 Office Action Response at 12 (emphasis added).) In arguing the distinction between the claims and Hecht, Kaufman argued that the claims involve "extracting already existing relationships from the database" and "traversing existing relationships across

tables within a database so as to navigate the database *structure*." (*Id.* at 12-13 (emphasis added).)

These portions of the '981 Patent specification and the file history make clear that "wherein said relational database may be of any arbitrary size or complexity" refers to the schema of the database, rather than, *e.g.*, the amount of data within the database.

2. "automatically generating" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
The "generating" - comprising steps (a), (b), and (c) - is carried out upon being triggered by a user, without requiring further intervention by the user in order to complete the generation of the UI.	Plain and ordinary meaning

As with term 1, "plain and ordinary meaning" (as proposed by Microsoft) is not far off. Plaintiff has proposed a construction for this term, to make it explicit that "automatic" means that the specific claim steps recited all take place "automatically" after being triggered by the user.

The specification supports this interpretation in the "Field of the Invention" section, which states that the patent relates to "automatically generating without any custom programming a user interface for the database, and/or a complete application utilizing the database." ('981 Patent at 2:22-26.) In the detailed description, the specification makes clear that "Software . . . automatically and dynamically ("on-the-fly") generates a fully functional UI system . . . based upon, and connected directly to, the underlying data model." ('981 Patent at 11:23-28.)

In distinguishing other systems, the specification makes clear that "the construction of front-end [database] applications is generally undertaken using conventional . . . computer

languages, which require by-hand coding at a very low level of functionality." ('981 Patent at 2:53-56.) Then-available tools for easing development were "limited to fairly specific (and, still, fairly low-level) uses—among them, providing more-sophisticated or 'richer' controls for manipulating individual data elements" and sometimes "offering 'form generator' or 'wizard' facilities to automatically generate the code for a simple UI display which manipulates a single underlying (back-end) data table." ('981 Patent at 2:57-64.) "Even with such tools, considerable work remain[ed] in building a complete, fully-functional UI for a back-end schema of any appreciable size or complexity." ('981 Patent at 2:65-67.) And although back-end and front-end components are closely synchronized, the respective development efforts are generally separate, "with the requisite synchronization and parallels in structuring being effected only manually." ('981 Patent at 2:50-51.)

"Automatically generating" means that steps (a), (b), and (c) are carried out upon being triggered by a user, without requiring further intervention by the user in order to complete the generation of the UI.

3. "defining a user interface paradigm comprising a set of modes for interacting with a given database table, said modes comprising create, retrieve, update and delete, and a corresponding display format for each mode" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
For each of the specified modes of operation (create, retrieve, update and delete), the generated UI includes among its provided display formats at least one display format which supports that operation.	defining a set of distinct user interfaces comprising a screen or window for each of the operations of creating, retrieving, updating, and deleting data in a given database table

For this construction, Microsoft has inserted the word "distinct" into the claim language to limit the scope of the claims. Microsoft thereby seeks to add a limitation to the claim language

that would create the situation in which any combination of two operations in any single template (display format) would avoid infringement. This would be improper.

The claim language is "a user interface paradigm comprising a set of modes . . . comprising create, retrieve, update and delete, and a corresponding display format for each mode." The word "distinct" does not appear in this language, and including a "distinct" requirement would go against the claim language itself, as well as the disclosure in the specification.

The language recites that there is a "corresponding" display format for each mode. Corresponding can be a one-to-one relationship, or a one-to-many. With a list of states and another list of state capitals for the United States, each state would have one (and only one) corresponding capital. But, in a book index, each indexed term would have one or more corresponding pages, and multiple indexed terms could (and generally would) appear on any single page.

The specification makes clear that the delete mode, in particular, can be added to the reference implementation "simply by adding (according to the user's access rights, potentially) another pushbutton within the Edit-mode display." ('981 Patent at 5:63-6:3.) Requiring each mode to have a *distinct* display, as Microsoft urges, would exclude the preferred embodiment from the scope of the claims. This would be counter to an established canon of claim construction as noted above.[4]

---

[4] Indeed, Microsoft's motivation in proposing this construction is clear from the fact that the accused implementation in Microsoft's Dynamic Data product handles "Delete" in the same manner as described in the '981 Patent's specification – by adding a "Delete" control to the edit/update screen.

The specification shows that each of the denoted operations should be supported by the complement of mode displays. "The generated UI comprises all mode displays (e.g., browse, search, edit, and add) for all tables, and a full complement of mechanisms, integrated into the mode displays for representing, navigating, and managing relationships across tables." ('981 Patent at 3:43-47; 12:24-27.) However, there is no requirement that each mode is represented on a "distinct" display.

Microsoft's proposed construction is further flawed by requiring a single "screen or window" for each of the CRUD modes. The specification makes clear that there can be different types of screen displays for any given mode. The specification provides an example of one list of modes that the user may navigate among: "FULL BROWSE, FILTERED BROWSE, NEW SEARCH, REVISED SEARCH, and ADD." ('981 Patent at 5:27-41.) This example includes multiple browses and multiple searches (each corresponding to "Retrieve" in CRUD). Under Microsoft's proposed construction, the preferred embodiment, which provides multiple retrieve mode display options, would be excluded from the scope of the claims.

In sum, the claims do not say that the display formats are mutually "distinct," and there is no basis in the claims or specification, and nothing from the prosecution history, to support inserting this additional requirement into the claims to limit the claim scope.

The claims require that, for each of the specified modes of operation (create, retrieve, update and delete), the generated UI includes among its provided display formats at least one display format which supports that operation.

4. "integrates into each said mode display processes for representing, navigating, and managing said relationships across tables, for selecting among said modes, and for navigating across said tables and interacting in accordance the selected mode with the data in the tables that are reached by said navigation, while observing and enforcing relational interdependencies among data across said tables" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
Integrates into each said mode display one or more processes for representing, navigating, and managing said relationships across tables, selecting among said modes for tables reached by said navigation, and interacting in accordance with the selected mode with the data in the tables that are reached by said navigation, while observing and enforcing relational interdependencies among data across said tables. The mode displays, taken together, provide the full complement of such processes.	integrates into each distinct user interface comprising a screen or window processes for (1) representing, (2) navigating, and (3) defining the relationship between one table and another; for selecting among the user interfaces; and for navigating across said tables and interacting according with the selected user interface with the data in the tables that are reached by said navigation, while observing and enforcing relational interdependencies among data across said tables

As illustrated in the specification and discussed above, the generated client application provides a set of user interface elements (mode displays) that supports all of the specified modes of interacting with the database data. Further, each mode display integrates processes relevant in the context of that display for permitting navigation from table to related table and working with the tables reached by such navigation, and the set of display modes, taken together, provide the full complement of such processes. All of this is done while preserving relational interdependencies among data across tables, which is also known as "referential integrity."

Microsoft, for its part, doubles down on the "distinct" user interfaces limitation it seeks to insert in the claim language, arguing further that the claim language requires that each such "distinct" user interface incorporate all "processes" to be supported by the entire application.

This interpretation is not at all dictated by the claim language, is at variance with the specification, and would read on no embodiment described in the specification.

The specification states that "the generated UI comprises all mode displays for all tables, with integrated . . . mechanisms for representing, navigating, and managing relationships across tables." ('981 Patent at 3:43-47; 12:24-27.) It is clear from this disclosure that all of the claimed mechanisms are implemented among the full complement of screen displays provided by the generated application. However, this does not assert that each screen display implements each and every such mechanism. Microsoft's proposed construction to that effect would add limitations to the claims that do not exist.

Microsoft's proposed construction would not read on the embodiments disclosed in the specification. The screen examples shown in the specification, taken together, reflect each of the processes specified in the subject claim language (as individually addressed below). But not all screens implement all mechanisms. As one example, the "Retrieve" mode displays of the specification, Figs. 1 and 2, provide for representing and navigating relationships across tables, and selecting among modes, but not for "managing said relationships across tables." This makes sense, because in a read-only mode, such as a "Retrieve" mode, there is nothing that needs "managing." To read Microsoft's further limitations into the claims would require mindless replication for no purpose and contrary to the examples and logic of the specification.[5]

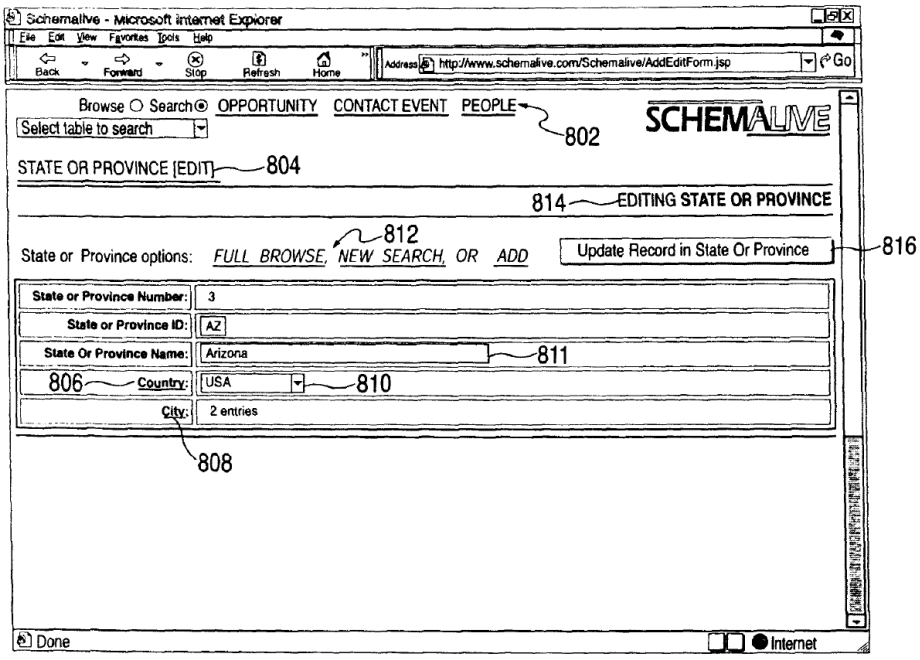
[5] Again, Microsoft's own implementation in Dynamic Data follows the same pattern as that disclosed in the specification – the retrieve screen similarly does not provide controls for managing relationships across tables, which would be pointless in that context.

5. "representing . . . said relationships across tables" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
Visually representing at least the existence of a relationship from the data record(s) in the table being viewed to the corresponding data in another table (if any) that is related to the viewed-table data record(s), according to the data model for the database.	Plain and ordinary meaning, consistent with the above construction, i.e., "representing . . . the relationship between one table and another"

For each relationship between two tables, the application generated in accordance with the Asserted Claims visually represents relationships between the *data records* in those tables. The specification illustrates at least three mechanisms for visually representing such relationships: hyperlinks, dropdowns, and substitutions of meaningful words for what are typically numerical foreign key values.

The use of hyperlinks to visually represent data relationships across tables is illustrated, for example, in the "Country" link, 806, in Fig. 8 of the '981 Patent:



In this display, clicking on the word "Country" takes the user to the detailed "USA" record from the separate Country table. The link 806 therefore visually represents a relationship between the displayed Arizona data record in the State or Province Table and the related USA data record in the Country table.

Another type of relationship visually represented by a hyperlink is a "MASTER/DETAIL" relationship (a.k.a. "parent/child" or "one-to-many") relationship, in which "multiple foreign-key table records are referenced by single primary-table record." ('981 Patent at 6:42-44.) These relationships can similarly be represented with a clickable link. Referring again to Fig. 8 above, a "drill-down" link, 808, is provided to a "detail" table, in this case, the "City" table. ('981 Patent at 5:52-53.) Clicking this link (from the State or Province "master" record shown in Fig. 8) would bring up a corresponding view of the detail table (in this case the City table) constrained in scope according to the master-table context (Arizona) so as to show only the two cities that exist in the database for Arizona. Again, the "City" link 808 represents a relationship between the state of Arizona as contained in the State or Province table, and the cities in Arizona that are included in the City table.

Similarly, the generated user interface illustrated in the specification uses dropdowns to visually represent data relationships across tables. The dropdown control 810 in Fig. 8 is an example. The value visible in Fig. 8 is "USA," one of the list of values shown when the dropdown control is opened. This list is populated from, and represents, the records in the Country table. This visually represents the possible values that may be chosen from the foreign table for association with the current primary-table record.

Furthermore, the generated user interface uses a technique of substituting human-friendly descriptors derived from a foreign table for numerical foreign-key cross-references maintained in

the primary table. This is another method that the specification describes for visually representing the relationships between data elements across tables. In the Department/Employee example given previously, the cross-reference for Marie Curie's department is "200" (the Department_ID) rather than "Chemistry" (the Department Name). The generated UI substitutes the name for the number in order to make the cross-reference more readable and human-friendly. The user sees the "Chemistry" department, instead of department "200."

An example of such substitution can be seen in Fig. 8. The Country dropdown represents a reference to a foreign table, *i.e.*, the Country table. In raw form, the references in the dropdown would be to the "Country_ID" field-values within the Country table (which are typically numbers). Instead, the generated user interface replaces these numbers with meaningful words, in this case the country name (or abbreviation), for display in the dropdown. A similar mechanism is employed in rendering column values (within a "browse" display) that are foreign keys to another table, for example the "Modified by Users" column in Fig. 9 of the '981 patent, which lists user names rather than ID numbers.

Thus, when viewed in light of the specification, "representing . . . relationships across tables" means visually representing at least the existence of a relationship from each data record in the table being viewed (the "viewed table") to the corresponding record in another table (if any), according to the data model for the database.

6. "navigating . . . said relationships across tables" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
For the visually represented cross-table relationships, navigating from the data in the viewed table to a display of the data in a related table that is specifically related to the viewed-table data record.	Plain and ordinary meaning, consistent with the above construction, i.e., "navigating . . . the relationship between one table and another"

The construction of navigating the relationships follows closely from the discussion of representing the relationships. As outlined above, two types of relationships described in the specification are foreign key relationships and master-detail relationships, and those relationships can be represented via links, dropdowns, and/or substitutions.

Links allow for navigation from one table to another table across the represented relationships. (*See* '981 Patent at 5:50-53.) As discussed further in the specification:

> Traversing relationships (either cross-reference or master/detail) is referred to as "drill-down" . . . e.g., in the embodiment shown in FIG. 9B, a click on a "drill-down" button 904 (shown in this example as "State or Province") allows for a "drill-down" to related detail records (shown in FIG. 9C) . . . Drill-downs are supported by enabling "hot-linked" (or "clickable") labels for the related data fields in the primary table (stack context) (see FIGS. 9B and 9C)

('981 Patent at 9:23-30.)

This passage, in combination with the claim language, makes clear that navigating relationships requires navigating from data in the viewed table to a display of data in a related table that is specifically related to the viewed-table record.

7. "managing . . . said relationships across tables" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
Managing the relationships of records in a given table with corresponding records in a related table, for example, by way of a dropdown that limits selection of an added or edited value for a record in the given table to the permissible values as exist within the records of the related table.	Plain and ordinary meaning, consistent with the above construction, i.e., "defining the relationship between one table and another" or "creating, revising, or manipulating a definition that specifies how one table is related to another"

Microsoft's proposed definition for this term is incorrect and would entirely distort the claim language. Microsoft essentially argues that "managing" is the same as "defining" and/or "creating" or "changing."

As is clear from the claim language, to create the end-user interface, a database is scanned for its data model and, from the scan, a client application is created, with the above-described capabilities. One of the objectives listed in the summary of the patent is to provide this application "once a back-end schema has been designed and constructed." ('981 Patent at 3:13-25)

Defining, creating, revising, and/or manipulating relationships between tables, as Microsoft is arguing in its two different constructions, would require that the constructed application have the ability to go back and revise the schema on which it was based. This is not the intent of the claims, and contrary to the entire disclosure of the specification.

"Managing" means managing the data affecting relationships between individual table records. When the schema defines relationships between tables, the application allows the manipulation of *data* within the database (adding or editing data) so as to alter the *connections* between *specific records* across the related tables. The primary example discussed in the

specification is restricting input for foreign-key fields via a dropdown list populated with data from the foreign table.

The specification includes "CROSS-REFERENCE FIELD 810 to generate dropdown lists of available foreign-key values (with automatic correlation to display-name labels)." ('981 Patent at 5:54-56.) As discussed above, the dropdown is populated with data from the corresponding field of the referenced foreign table. This mechanism constrains the user to pick data from only the dropdown, preventing the user from entering data that does not exist in the linked foreign table, and thereby preserving referential integrity. For instance, when adding an employee in the Employee/Department example above, only existing departments are shown. This prevents the user from adding an employee with a non-existent department.

Kaufman's proposed construction is supported by the prosecution history. As outlined in the prosecution history of the parent application,

> By way of concrete example, a "client application" as contemplated by the present invention would allow a user to add a new employee to the Employee table in Hecht, and in so doing, restrict the user so that the Department that could be entered for the new employee would have to be one of the Departments defined in the Department table (because of the "foreign key" relation between the two tables).

(U.S. Patent Application No. 10/428,209, December 12, 2006 Office Action Response at 8.)

The claim language, when read in light of the specification and the prosecution history, requires managing relationships to mean "Managing the relationships of records in a given table with corresponding records in a related table, for example, by way of a dropdown that limits selection of an added or edited value for a record in the given table to the permissible values as exist within the records of the related table."

8. "while observing and enforcing relational interdependencies among data across said tables" (All Asserted Claims)

Plaintiff's Proposed Construction	Microsoft's Proposed Construction
All stated navigation, management, and interaction is carried out consistent with, and preventing violation of, all cross-table interdependencies, e.g., so as to preserve the referential integrity of the relational database.	Plain and ordinary meaning.

Again, the plain meaning of this term is not far off, but to prevent confusion, Kaufman is requesting a construction. The language of the claims requires that the above-discussed operations (navigation, management, and interaction) are carried out while "observing and enforcing relational interdependencies among data across said tables."

The specification states

> It is yet a further object of the invention that the application so presented reveals (and enforces) the relational/hierarchical organization among the tables within the back-end via smoothly integrated UI mechanisms which are embedded directly into the base-table screen displays—providing a natural, powerful, and easy-to-use environment for managing complex data relationships and interactions.

('981 Patent at 3:26-32.) This passage, as well as the remaining claim language, as discussed throughout this paper, make clear that the claims require that the operations be carried out "consistent with, and preventing violation of, all cross-table interdependencies, e.g., so as to preserve the referential integrity of the relational database." This is the result of the above functionality, of representing, and especially of managing relationships while data is manipulated. By doing so, referential integrity is preserved.

Dated: November 8, 2016

/s/ Ronald Abramson

Ronald Abramson
Tel: (212) 822-0163
Email: ronald.abramson@lewisbaach.com
David G. Liston
Ari J. Jaffess
Alex G. Patchen
LEWIS BAACH pllc KAUFMANN MIDDLEMISS
The Chrysler Building
405 Lexington Avenue, 62nd Floor
New York, New York 10174

Attorneys for Plaintiff
Michael Philip Kaufman